Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A system for removing gaps from streams of packets, comprising:

a packet splitter configured to:

receive the packets, each of the packets including a packet header and packet data, and

separate the packet header from the packet data for each of the packets;

a header buffer configured to store the packet headers;

a data buffer configured to store the packet data; and

a packet combiner configured to:

reassemble the packets from the packet headers in the header buffer and the packet data in the data buffer, the reassembling causing gaps to occur between the packet headers and the packet data within the reassembled packets, and

remove the gaps from the reassembled packets.

2. (original) The system of claim 1, wherein the packet splitter is further configured to:

identify a number of bytes of packet data stored in the data buffer, and

provide the identified number of bytes to the header buffer.

3. (original) The system of claim 1, wherein the streams include packet streams of different bandwidths.

4. (original) The system of claim 1, wherein the packet combiner includes:

a multiplexer configured to multiplex the reassembled packets, and

a byte packer configured to remove the gaps from the multiplexed packets.

5. (original) The system of claim 1, wherein the packet combiner includes:

a stream map configured to output stream identifiers corresponding to the streams, and

handshake logic configured to:

read the packet headers from the header buffer and the packet data from the data buffer based on the stream identifiers, and

reassemble the packets using the packet headers and the packet data.

6. (original) The system of claim 1, wherein the packet combiner includes:

a memory configured to:

store the reassembled packets based on write stream identifiers from a stream map, and

output the reassembled packets based on read stream identifiers from the stream map.

7. (original) The system of claim 6, wherein the memory includes:

a buffer that includes a plurality of blocks corresponding to the streams,

a write pointer register configured to store a plurality of the write stream identifiers, and

a read pointer register configured to store a plurality of the read stream identifiers.

8. (original) The system of claim 6, wherein the packet combiner further includes:

a first multiplexer configured to multiplex one or more of the reassembled packets output by the memory based on the write or read stream identifiers, and

a second multiplexer configured to multiplex one or more of the reassembled packets output by the memory based on the write or read stream identifiers.

9. (original) The system of claim 8, wherein the first multiplexer includes a 2:1 multiplexer and the second multiplexer includes a 3:1 multiplexer.

10. (original) The system of claim 8, wherein the first multiplexer includes a 3:2 multiplexer and the second multiplexer includes a 2:1 multiplexer.

11. (original) The system of claim 8, wherein the packet combiner further includes:

a first byte packer configured to remove gaps from the multiplexed packets from the first multiplexer, and

a second byte packer configured to remove gaps from the multiplexed packets from the second multiplexer.

12. (original) The system of claim 8, wherein the packet combiner further includes:

a third multiplexer configured to multiplex the reassembled packets, and

a byte packer configured to:

remove gaps from the multiplexed packets from the third multiplexer to create gap-free packets, and

store the gap-free packets in the memory.

13. (original) The system of claim 1, wherein the packet combiner includes:

handshake logic configured to:

read the packet headers from the header buffer and the packet data from the data buffer, and

reassemble the packets from the packet headers and the packet data,

a first multiplexer configured to multiplex the reassembled packets from the handshake logic,

a byte packer configured to remove gaps from the multiplexed packets from the first multiplexer to generate gap-free packets,

a memory configured to temporarily buffer the gap-free packets,

a second multiplexer configured to multiplex one or more of the gap-free packets from the memory, and

a third multiplexer configured to multiplex one or more of the gap-free packets from the memory.

14. (original) The system of claim 13, wherein the first multiplexer includes a 3:2 multiplexer, the second multiplexer includes a 2:1 multiplexer, and the third multiplexer includes a 3:1 multiplexer.

15. (original) The system of claim 1, wherein the packet combiner further includes:

a stream map configured to output stream identifiers corresponding to the streams, the stream identifiers controlling processing of the packets by the packet combiner.

16. (currently amended) A system for removing gaps from ~~streams of~~ packets, comprising:

means for receiving ~~the packets, each of the packets~~ a packet including a packet header and packet data;

means for separating the packet header from the packet data for ~~each of~~ the ~~packets~~ packet;

means for separately storing the packet ~~headers~~ header and the packet data;

means for reassembling the ~~packets using~~ packet based on the stored packet ~~headers~~ header and the stored packet data, the reassembling causing a gap to occur within the packet between the packet header and the packet data; and

means for removing ~~gaps~~ the gap from the reassembled ~~packets~~ packet.

17. (currently amended) A method for removing gaps from streams of packets, comprising:

obtaining the streams of packets, each of the packets including a packet header and packet data;

separating the packet header from the packet data for each of the packets;

separately storing the packet headers and the packet data;

reassembling the packets using the stored packet headers and the stored packet data, the reassembling causing gaps to occur within the reassembled packets between the packet headers and the packet data; and

removing the gaps from the reassembled packets.

18. (original) The method of claim 17, wherein the removing gaps includes:

multiplexing the reassembled packets based on a stream identifier, and

removing the gaps from the multiplexed packets using a byte packing technique.

19. (original) The method of claim 17, wherein the reassembling includes:

receiving stream identifiers corresponding to the streams,

reading the packet headers and the packet data based on the stream identifiers, and

reconstructing the packets from the packet headers and the packet data.

20. (original) The method of claim 17, further comprising:

storing the reassembled packets in a memory based on write stream identifiers corresponding to the streams; and

outputting the reassembled packets from the memory based on read stream identifiers corresponding to the streams.

21. (original) The method of claim 20, wherein the storing the reassembled packets includes:

writing each of the reassembled packets to a block of the memory that corresponds to the stream to which the reassembled packet belongs.

22. (original) The method of claim 17, further comprising:

multiplexing one or more of the reassembled packets by a first multiplexer based on stream identifiers corresponding to the streams; and

multiplexing one or more of the reassembled packets by a second multiplexer based on stream identifiers corresponding to the streams.

23. (original) The method of claim 22, further comprising:

subjecting the multiplexed packets from the first multiplexer to a first byte packing process; and

subjecting the multiplexed packets from the second multiplexer to a second byte packing process.

24. (original) The method of claim 17, wherein the removing gaps includes:

subjecting the reassembled packets to a byte packing process to create gap-free packets.

25. (original) A system that operates in first and second modes to remove gaps from streams of data, comprising:

handshake logic configured to:

obtain a header portion and a body portion of a data unit corresponding to one of the data streams, and

reassemble the data stream from the header and body portions;

a first multiplexer configured to multiplex the reassembled data stream from the handshake logic;

a byte packer configured to remove gaps from data units within the multiplexed data stream to generate a gap-free data stream;

a memory configured to temporarily buffer the gap-free data stream;

a second multiplexer configured to multiplex the gap-free data stream in the first mode; and

a third multiplexer configured to multiplex the gap-free data stream in the second mode.

26. (original) The system of claim 25, wherein the first multiplexer includes a 3:2 multiplexer, the second multiplexer includes a 2:1 multiplexer, and the third multiplexer includes a 3:1 multiplexer.

27. (original) The system of claim 25, further comprising:

a stream map configured to output stream identifiers corresponding to the data streams, the stream identifiers controlling processing of the data streams by the system.

28. (original) A system that operates in first and second modes to remove gaps from streams of data, comprising:

means for obtaining a header portion and a body portion of a data unit corresponding to one of the data streams;

means for reassembling the data stream from the header and body portions;

means for multiplexing the reassembled data stream by a first multiplexer;

means for removing gaps from data units of the multiplexed data stream from the first multiplexer to generate a gap-free data stream;

means for multiplexing the gap-free data stream by a second multiplexer in the first mode; and

means for multiplexing the gap-free data stream by a third multiplexer in the second mode.

29. (original) A method for removing gaps from streams of data in first and second modes, comprising:

receiving a header portion and a body portion corresponding to one of the data streams;

reassembling the data stream from the header and body portions;

multiplexing the reassembled data stream by a first multiplexer;

removing gaps from the multiplexed data stream from the first multiplexer to generate a gap-free data stream;

multiplexing the gap-free data stream by a second multiplexer in the first mode; and

multiplexing the gap-free data stream by a third multiplexer in the second mode.

30. (original) A network device, comprising:

a switching fabric; and

a plurality of packet processors connected to the switching fabric, each of the packet processors including:

a stream map configured to provide stream identifiers corresponding to a plurality of streams of packets;

a first multiplexer configured to multiplex packets based on the stream identifiers;

a byte packer configured to remove gaps from the multiplexed packets from the first multiplexer to generate gap-free packets;

a memory configured to store and output the gap-free packets based on the stream identifiers;

a second multiplexer configured to multiplex one or more of the gap-free packets based on the stream identifiers; and

a third multiplexer configured to multiplex one or more of the gap-free packets based on the stream identifiers.

31. (original) A network device, comprising:

a switching fabric; and

a plurality of packet processors connected to the switching fabric, each of the packet processors including:

a stream map configured to provide stream identifiers corresponding to a plurality of streams of packets;

a memory configured to store and output packets based on the stream identifiers;

a first multiplexer configured to multiplex one or more of the stored packets based on the stream identifiers;

a first byte packer configured to remove gaps from the multiplexed packets from the first multiplexer to generate gap-free packets;

a second multiplexer configured to multiplex one or more of the stored packets based on the stream identifiers; and

a second byte packer configured to remove gaps from the multiplexed packets from the second multiplexer to generate gap-free packets.